Algoritmos y Estructuras de Datos III - 1er Parcial
Fecha de examen: 29-ABR-2022

	Nº Orden	Apellido y nombre		L.U.	# hojas [1]
Notas:	Ej1	Ej2	Ej3	Ej4	Final

Aclaraciones: El parcial se aprueba con 2 (dos) ejercicios aprobados. Cada hoja debe estar numerada y debe tener el número de orden y L.U. El parcial dura 4 horas y es a libro abierto.

1) Se arrojan simultáneamente n dados, cada uno con k caras numeradas de 1 a k. Queremos calcular todas las maneras posibles de conseguir la suma total $s \in \mathbb{N}$ con una sola tirada. Consideramos que los dados son distinguibles, es decir que si $n = 2$ y $k = 2$, entonces existen 2 posibilidades que suman $s = 3$: 1 en el primer dado y 2 en el segundo y viceversa.

   a) Definir en forma recursiva la función $f\colon \mathbb{N}^2 \to \mathbb{N}$ tal que $f(n, s)$ devuelve la cantidad de formas de conseguir la suma s con n dados de k caras.

   b) Demostrar que f tiene la propiedad de superposición de subproblemas.

   c) Definir un algoritmo *top-down* para calcular $f(n, s)$ indicando claramente las estructuras de datos utilizadas y la complejidad resultante.

   d) Escribir el (pseudo-)código del algoritmo top-down resultante.

   Aprobación: definir y justificar correctamente f, indicando cómo se computa $f(n, s)$ en tiempo $O(nk \operatorname{mín}\{s, nk\})$.

2) Un grafo *etiquetado* G es un grafo cuyos vértices tienen cada uno una etiqueta distinta en el conjunto $\{1, \ldots, n\}$, donde n es la cantidad de vértices de G. En G, cada arista se etiqueta utilizando las etiquetas de sus vértices incidentes. Dos grafos etiquetados de n vértices son *iguales* cuando tienen el mismo conjunto de aristas etiquetadas. Una *orientacion acíclica* de un grafo (etiquetado o no) G es un grafo orientado H que no tiene ciclos dirigidos y que resulta de asignarle una dirección a cada arista de G. Ciertamente, si G es etiquetado, entonces H mantiene las mismas etiquetas que G. Probar las siguientes afirmaciones:

   a) Todo grafo etiquetado con al menos una arista tiene una cantidad par de orientaciones acíclicas distintas.

   b) Toda orientación acíclica de un grafo tiene al menos un *sumidero* (es decir un vértice con grado de salida cero).

   c) Toda orientación acíclica de K_n (el grafo completo de n vértices) tiene un único sumidero.

   d) Si a una orientación acíclica H de un grafo le agregamos un nuevo vértice v junto a una arista $u \to v$ para todo vértice u de H, entonces el grafo resultante es la orientación acíclica de un grafo.

   e) El grafo K_n etiquetado tiene exactamente $n!$ orientaciones acíclicas distintas.

   Aprobación: demostrar correctamente al menos 3 de los enunciados propuestos.

3) Dada una matriz de valores enteros de tamaño $m \times n$, $M^0 \in \mathbb{Z}^{m\times n}$, se define el siguiente proceso iterativo (ver Figura 1):

$$M_{ij}^{t+1} = \operatorname{máx}\left\{ M_{kl}^{t} \;\middle|\; \begin{array}{l} \operatorname{máx}(1, i-1) \leq k \leq \operatorname{mín}(m, i+1) \\ \operatorname{máx}(1, j-1) \leq l \leq \operatorname{mín}(n, j+1) \end{array} \right\}$$

   Es decir que en cada paso temporal se calcula, para cada elemento de la matriz, un nuevo valor igual al valor máximo entre sus vecinos inmediatos (en el paso temporal anterior). Notar que la casilla en cuestión también se considera a la hora de calcular el valor máximo.

[1] Incluyendo esta hoja.

$$\begin{pmatrix} 1 & 5 & 10 & 5 \\ 6 & 4 & 12 & 4 \\ 10 & 5 & 12 & 11 \\ 5 & 11 & 23 & 9 \end{pmatrix} \qquad \begin{pmatrix} 1 & 5 & 10 & 5 \\ 6 & 4 & 12 & 4 \\ 10 & 5 & 12 & 11 \\ 5 & 11 & 23 & 9 \end{pmatrix}$$

Figura 1: Dos ejemplos de vecindarios a considerar al actualizar los valores indicados con círculos

a) Observar que este proceso converge en una cantidad finita de pasos, i.e., existe $t \geq 0$ tal que $M^{t+1} = M^t$. Dar una justifiación.

b) Diseñar un algoritmo que encuentre la mínima cantidad de pasos temporales necesaria para converger, explicando claramente su implementación y por qué es correcto.

Aprobación: el algoritmo propuesto debe tener complejidad $O(mn)$ y su corrección debe estar bien justificada.

4) Dado un grafo pesado y conexo G y un valor $k \in \mathbb{N}$, queremos construir un grafo H que tenga un árbol generador cuya arista de peso máximo tenga peso exactamente k. Para construir H podemos cambiar los pesos de las aristas de G. El costo de cambiar el peso $w_G(e)$ que una arista e tiene en G por el peso $w_H(e)$ que e tiene en H es $|w_H(e) - w_G(e)|$. Por lo tanto, el costo de construir H es

$$\sum_{e \in E(G)} |w_H(e) - w_G(e)|.$$

Tenemos a nuestra disposición un algoritmo que puede construir un AGM de un grafo con m aristas en $O(m\alpha^{-1}(m))$ tiempo. Diseñar un algoritmo de tiempo $O(m\alpha^{-1}(m))$ que, dados G y k, encuentre el grafo H cuyo costo de construcción sea mínimo.

Ayuda: Resolver el problema suponiendo que todo AGM tiene una arista de peso al menos k. En el caso que la suposición sea errónea, construya H realizando "pocos" cambios sobre G.

Aprobación: el algoritmo propuesto debe tener la complejidad pedida y su corrección debe estar bien justificada.